LINE 封包特徵分析預測使用者網路活動

陳詰昌
警政署刑事警察局偵查第九大隊
jay@email.cib.gov.tw

摘要

過去網路犯罪偵查，常對涉嫌對象進行通訊監察蒐證，由於隱私權及安全性意識，幾乎所有網路通訊均以 SSL/TLS 或其他方式進行加密傳送,對於釐清犯罪事實造成相當大衝擊。在智慧型手機普及化後，即時通訊軟體已漸漸取代傳統話務，通訊種類愈來愈多元，不但可以進行語音視訊通話、文字訊息傳送，亦可傳送圖片、影像及各類檔案。本文針對通訊軟體 LINE 為主題，以深度封包檢測概念應用於智慧型手機加密封包，首先以簡易方式取出 LINE 傳送之封包後，去除掉無效封包，下一步再針對使用者網路行為常見之文字訊息傳送接收、語音通話、圖片影音傳送等行為進行分析，試圖找出各類網路行為之特徵，並將此特徵應用以 Wireshark 過濾語法呈現，使偵查人員易於分析及判斷通訊監察封包內容，指引偵辦方向。

關鍵詞：加密封包分析、LINE、通訊監察

Profile User Activity through LINE Encrypted Traffic

Jay-chang Chen[1*]
[1]Criminal Investigation Bureau, National Police Agency
[1]jay@email.cib.gov.tw

Abstract

Smart Phone and 4G Network are popular today. Many criminals use instant message application as a communication tools. LINE is the most popular instant message application in Taiwan. Telecommunication surveillance is ineffective for encrypted traffic. We setup a control environment to capture the traffic of smart phone, and filter all the packets related to LINE. It is helpful for law enforcement to extract some information from encrypted traffic.

Keywords: Encrypted Traffic, LINE, Telecommunication Surveillance

壹、 前言

1.1 研究背景與動機

過去數十年間，臺灣地區許多重大刑案偵辦過程，往往與通訊監察息息相關，但曾幾何時，這項偵辦刑案的屠龍刀突然不靈光。主要原因在於過去以語音話務與簡訊為主的年代，政府機關可以透過法令與強制手段由電信交換機房掌握特定對象通訊內容；但在 4G 行動網路通訊普及後，智慧型手機與即時通訊軟體(Instant Messaging Applications) 取代傳統電信話務，取而代之的是 Whatsapp、WeChat、Viber、LINE、Facebook 等以 SSL/TLS 等加密封包進行傳送的通訊模式，使偵查機關對此類新型態通訊模式束手無策。

隨著智慧型手機成長與 4G 行動網路普及，加密通訊問題愈來愈嚴重。根據調查台灣地區主流的通訊與社群軟體應該屬 LINE 與 Facebook 莫屬，以日本 LINE 公司 2016 年公布數據來說，LINE 在台灣地區用戶帳號超過 1,700 萬，在各年齡層滲透率幾乎都超過九成，再加上即時通訊軟體不但可傳送文字訊息，亦可進行語音通話、傳送影片、圖片及社群功能，LINE 幾乎取代傳統電信話務，使通訊監察光碟不再是語音內容，取而代之的是無法解密的封包。

對於傳統網路流量分析而言，對於未加密封包可進行通訊內容重建，使得通訊雙方所傳遞內容能夠一覽無遺，譬如過去 MSN、Yahoo Messenger、網頁聊天室等；但目前資訊安全與個人隱私至上的時代，幾乎所有通訊軟體或網頁瀏覽等都已採用 SSL/TLS 等方式進行加密[4]，儘管通訊監察能夠攔截雙方傳遞網路封包，順利解開雙方通訊內容仍具難度。但是隨著加密封包成為主流，雖然對通訊監察成為挑戰，但也陸續有學者提出針對加密封包進行分類與分析的研究，且將這樣的研究應用到智慧型手機網路封包分析上，由攔截智慧型手機封包來對使用者所使用應用程式進行分類與分析使用者行為。

1.2 研究目的

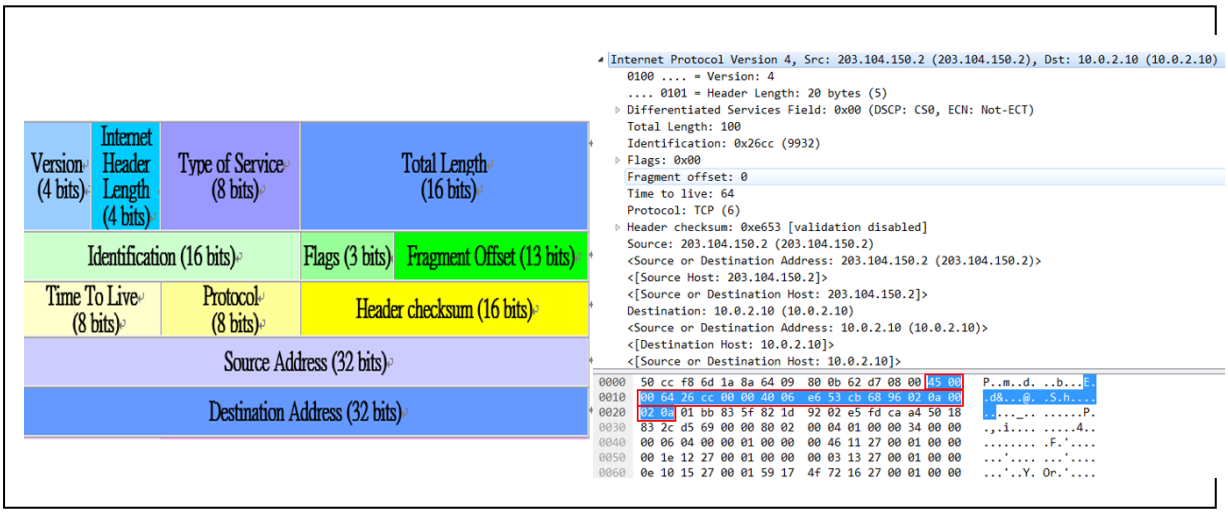
在目前臺灣法規授權範圍下，僅能對監察目標網路流量進行擷取，尚未授權可採中間人攻擊方式（MITM）偽造憑證[4]或以木馬程式入侵監察目標方式進行蒐證，本研究以現行通訊監察所得加密封包為研究基礎，嘗試對加密封包進行分類與分析。目前偵查人員普遍面臨之問題就是臺灣地區使用族群最多的通訊軟體 LINE，因此，本文採實驗方式模擬通訊監察擷取封包，以過去通訊監察實際案例進行驗證，將歸納結果提供予偵辦人員，使用於日後通訊監察封包分析之用。

貳、文獻探討

封包重組（Packet Reassembly）或深度封包檢測（deep packet inspection）已經有很多著作或論文，當然也有相當多成熟的工具可供使用者選擇，譬如 E-Detective、A-Packman、Wireshark、Xplico 等。由於科技發展與時代變遷，智慧型手機已成為民眾最常使用之電子通訊產品，因此通訊監察內容除語音通話外，又新增網路封包；但是目前監察系統並未能對網路封包進行深度分析，僅能針對未加密封包進行重組還原，對 LINE 使用者行為進行討論與研究。因此，就封包格式、封包檢測與歸類、通訊軟體加密等進行相關資料蒐集[1,8]。

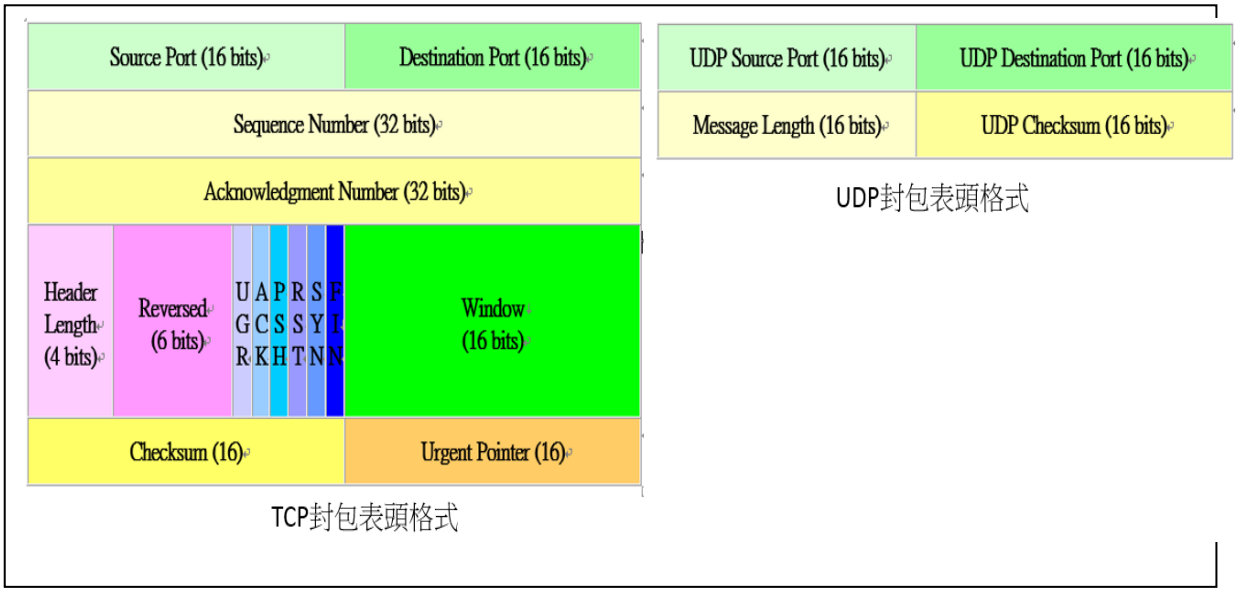
2.1 封包格式

封包是網路傳輸的資料單位，依據通訊協定每個封包含有 MAC 表頭、IP 表頭、TCP/UDP 表頭及實際資料等，在 IP 表頭格式及長度依序為版本次（IPv4 或 IPv6，長度 4bits）、標頭長度（Internet Header Length，長度 4bits）、服務類型（Type of Service，長度 8bits）、封包總長（Total Length，長度 16bits）、識別碼（Identification，長度 16bits）、標記（Flag，長度 3bits）、分割定位（Fragment Offset，長度 13bits）、延續時間（Time To Live，長度 8bits）、協定（Protocol，長度 8bits）、標頭檢驗值（Header Checksum，長度 16bits）、來源 IP 地址（Source IP address，長度 32bits） 、目的 IP 地址（Destination IP address，長度 32bits）等，詳如圖一。其中協定欄位值為 6（0x06）則為 TCP、值為 17（0x11）則為 UDP。



圖一：IP 表頭封包格式與 Wireshark 截錄封包資料欄位對照圖

依據 IP 表頭封包欄位-協定，將通訊協定分為 ICMP、TCP、UDP 等，本文以最常見的 TCP 與 UDP 兩大類為主，進行欄位及長度比較，詳如圖二。

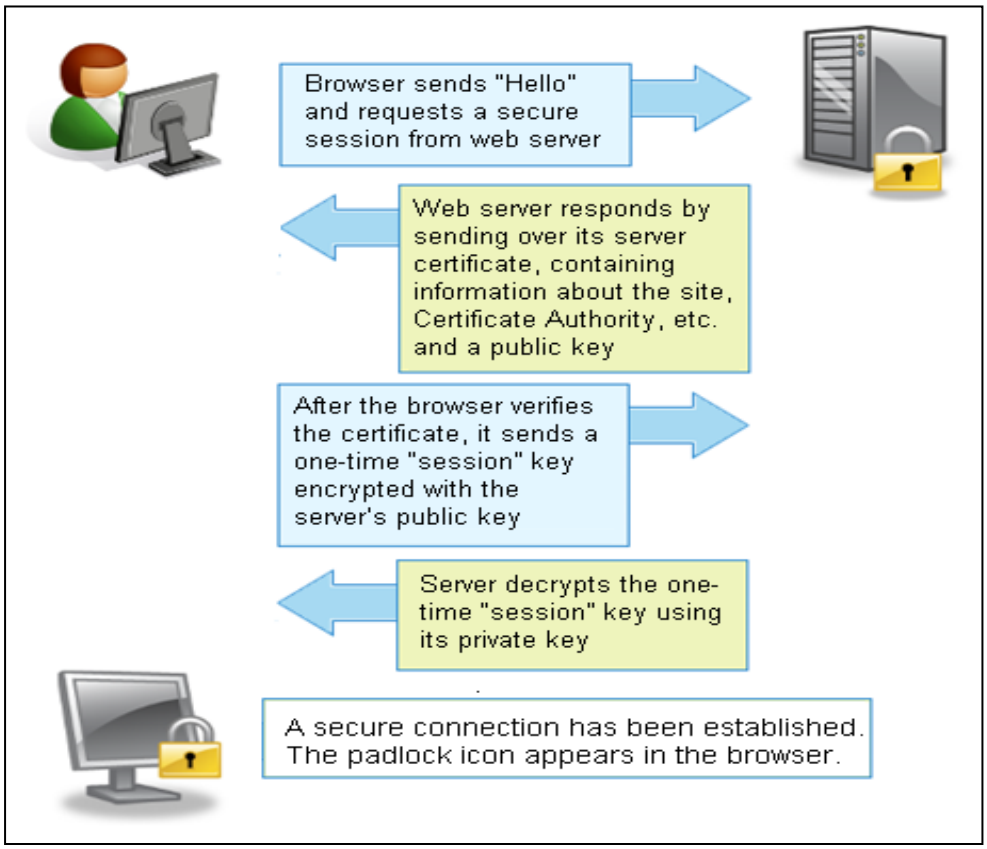


圖二：TCP 與 UDP 表頭格式對照

2.2 傳輸層安全協議(Transport Layer Security)/安全通訊協定（Secure Socket Layer ）

SSL 為 TLS 的前身，兩者都是一種網路安全協定用來做為資料加密、來源認證與完整性驗證的機制。安全通訊協定分為兩層，紀錄層與傳輸層。傳輸層安全協議利用非對稱加密演算來對通訊雙方進行身分認證，之後交換對稱金鑰作為會談金鑰（Session key），並用會談金鑰將通訊雙方交換的資料進行加密，確保通訊時的保密性和可靠性；紀錄層規定傳輸層資料封裝格式，目的是達到訊息的完整性及機密性要求[2]。

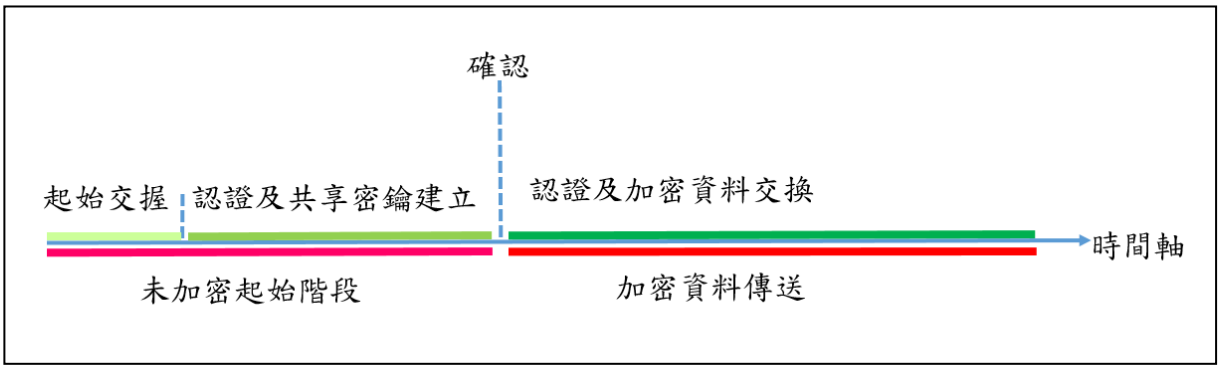
舉例來說，使用者使用安全通訊協定瀏覽伺服器網頁時，瀏覽器先送出請求給伺服器，告知伺服器本身可採用之演算法與相關資訊；接著伺服器送出這次通訊採用之演算法及伺服器本身憑證（含公鑰）；瀏覽器新增一秘密金鑰，並利用伺服器所傳來的公鑰進行加密，接著回傳加密後的秘密金鑰密文給伺服器；最後，伺服器使用私鑰解開瀏覽器傳來的秘密金鑰密文，取得秘密金鑰後，即利用此秘密金鑰來相互通訊。



圖三：SSL 交握過程

2.3 加密封包分類與分析

網路流量分析是網路安全項目之一，可以由網路流量偵測及分析網路攻擊、監測應用程式效能及偵測違反安全政策行為等。但是目前智慧型手機上常見通訊或社群軟體如 Signal、Whatsapp、Facebook、Google、iMessage、LINE、Wechat 多有進行封包加密機制，因此要將截錄所得封包進行分析就受到很大限制，但是仍然可以由封包流量取得一些資訊，可能來自未加密起始階段及加密傳送階段（圖四）。在 2015 年已經有研究人員提出針對智慧型手機加密封包進行分析，並能有效得知使用者在社群網路、電子郵件的活動行為；另外在洋蔥路由器所傳送的封包，雖然常見是 443 及 9010 通訊埠，但也已經有相關研究可由網路流量中辨識出使用洋蔥路由流覽之封包，這些都顯示加密網路流量並不是毫無用處；進一步來說，甚至已經有研究針對通訊軟體 KakaoTalk 之網路流量進行統計等分析，歸納出規則後，可以由加密封包流量瞭解使用者操作活動[9]。



圖四：網路安全協定

在流量分類上，分類依據可由流量負載（traffic payload）、流量特性（主機群聚性、主機、流向、封包特性）或綜合兩者與相關資訊等進行分類；而分類的方法可以使用深度封包檢測（Deep Packet Inspection）、圖形技術、統計方法及機器學習（Machine Learning）等進行分類。以深度封包檢測工具來說，在商業化工具有 PACE、NBAR（Cisco Network Based Application Recognition）等，另外開源工具有 nDPI、L7-filter 及 Libprotoident 等，各具有不同特色[7]。

2.4 WireShark

Wireshark 是一種可在多種平台使用的網路協議分析器，它可以用來截取、分析及過濾網路流量。這套軟體的優點是除了提供圖形化介面，讓使用者可以方便進行截取、分析外，也讓使用者透過的過濾語法（Berkeley Packet Filter）對截取封包進行多元分析方式。因此在過去數年間就有許多研究人員採用 Wiresahrk 作為攔截分析網路流量的工具，例如在 2013 年 Barghuthi 等人對社群網路即時通訊加密機制的研究論文也都是使用 Wiresahark 進行攔截分析[2]。

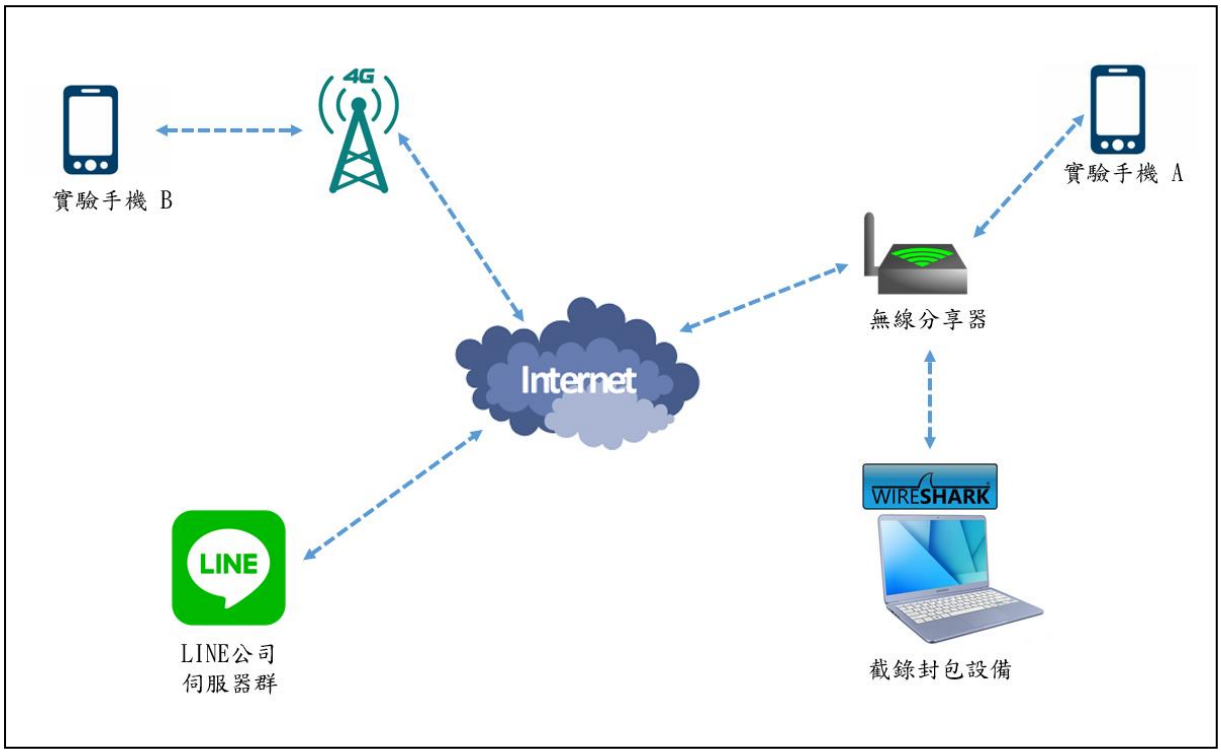
另外透過嵌入 MaxMind GeoIP 資料庫（http://dev.maxmind.com/geoip/legacy/geolite/）方式，在進行分析時可將 IP 位址所屬網路業者、自治區號碼等資訊一併帶出，亦或將其做為過濾語法，如 ip.geoip.asnum、ip.geoip.country 等都是衍生出過濾語法。

參、系統架構與結果

3.1 控制實驗

本文主要針對智慧型手機即時通訊軟體 LINE 封包進行分析，首先，我們建立控制

實驗環境用來模擬智慧型手機在網路通訊時，使用者使用 LINE 進行各種網路行為，並將一方假定為受監察端，將渠網路封包進行截錄，作為分析資料集。控制實驗環境建置如圖五，圖中實驗手機 A 為受監察端，以 Wireshark 軟體對無線分享器網路封包進行截錄，將實驗手機 A 上之 LINE 應用程式與 LINE 伺服器端或實驗手機 B 之間進行傳送封包進行截錄[8]。



圖五：控制實驗環境

3.2 特定應用程式封包過濾與分離

在智慧型手機封包中，針對各類應用程式封包分類與識別，有幾種方法：(一) 以通訊埠分類，但由於封包以 SSL 加密因素，許多封包傳遞方式改以 HTTP/HTTPS 等方式傳送，許多不同應用程式均使用 443 埠，因此以通訊埠進行分類作法難度大增。(二) 以 IP 與主機名稱分類，這種分類方式在 CDN（Content Delivery Network）及第三方網路服務興起後，IP 位址與主機名稱可能隨時會有變化而失效[5,6]。因此有學者提出以封包長度、傳送方向等變數，尋找各種應用程式特有模式與特徵，用來識別各類應用程式傳遞之封包，但這種方式可能因實驗設備、應用程式版本等因素差異，而造成實驗準確性下降。[4]

以本文研究對象即時通訊軟體 LINE 而言，LINE 應用程式除語音通話及視訊通話，可能出現與通訊方直接進行點對點通訊與封包傳遞情形，其他訊息傳送、圖片傳送、加退聊天室等服務均透過 LINE 伺服器達成，因此要由通訊監察截錄封包中，將 LINE 封

包過濾出來可透過簡單過濾語法即可達成。以 LINE 公司所屬網路而言，所使用之 IP 位址均屬於同一自治系統號碼（AS Number），而 LINE 所屬的自治系統號碼為 AS38631 及 AS23576，因此只要使用 Wireshark 以過濾語法「ip.geoip.asnum contains "AS38631"」即可將相關封包過濾顯示；另外接著濾除非必要封包[3]，所謂非必要封包就是確認（ack，過濾語法 not tcp.len==0）與重送信號封包（retransmission，過濾語法 not tcp.analysis.retransmission），確認信號封包特徵是負載（payload）長度為零，而重送信號封包則出現在封包損壞時，資料封包會被要求重送時出現，網路通訊在傳送大檔案會將檔案內容切為數十至數百個封包傳送，而 TCP 協定在封包傳送最大預設值為 1460 位元組，所以分析人員可將連續傳送 1460 位元組大小封包的行為視為一行為發生。

另外在分析封包時，如何將網路封包與使用者行為進行對應，在過去以有研究結果認為最適合進行分離封包與使用者行為之時間區間為 4.43 秒，因此本文亦採用此一最佳值套用於資料分析。

3.3 LINE 使用者行為種類

LINE 即時通訊軟體與其他同類應用程式一樣，大概有以下幾種主要功能：（1）傳送接收文字訊息（2）傳送接收圖片（3）傳送接收影片（4）傳送貼圖（5）加入/退出聊天室（6）加入/封鎖朋友（7）視訊/語音通話（8）其他。對於犯罪偵查來說，對偵查人員有幫助的應該是比較具機敏性的資訊，因此，本文著重於此部分使用者行為進行分析（表一）。

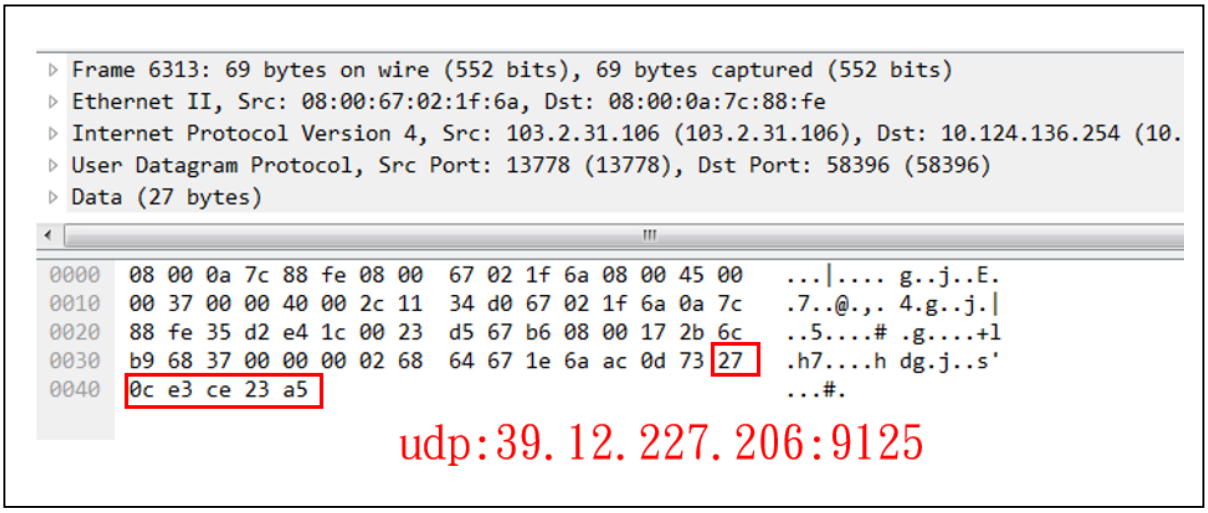
表一：LINE 使用者行為

使用者行為	功能描述
接收文字訊息	接收其他使用者傳送之訊息，如文字、圖像或貼圖
傳送文字訊息	傳送訊息予其他使用者，如文字、圖像或貼圖
加入聊天室	LINE 有成立群組功能，並可邀請好友加入群組
退出聊天室	使用者再加入群組後，可自主選擇是否退出群組
加入好友	將使用者加入好友清單
封鎖好友	封鎖特定使用者，在封鎖使用者後，遭封鎖方無法傳送訊息予另一方
檢視使用者資料	檢視其他使用者資料，如頭像及狀態
語音通話	與使用者進行語音通話
視訊通話	與使用者進行視訊通話

3.4 統計分析與特徵萃取

本文以前述實驗環境，執行特定 LINE 功能時將封包反覆進行截錄後，再以統計分析方法歸納每種使用者活動之特徵[8]；而在歸納此特徵前，LINE 有幾個重要的網域名稱 ga2.line.naver.jp、obs-tw.line-apps.com、obs-tw.rts.line.naver.jp、shop.line-cdn.net、obs.line-cdn.net、stickershop.line-cdn.net（貼圖，非屬 AS38631）等[10]，可以將之整理後，新增至 hosts 檔，以利後續 Wireshark 用於網域名稱解析用途。

(1) 視訊/語音通話：使用者 LINE 語音通話時，在響鈴階段開始，LINE 伺服器每隔 10 秒通過 11000 埠送出 UDP 封包，直到雙方通話結束時，即便未滿 10 秒，LINE 伺服器仍會再次通過 11000 埠送出 UDP 封包，因此由這特徵可以清楚掌握語音通話由響鈴開始至結束時間。另外通話時間超過一定時間後，雙方會由伺服器傳遞轉換為點對點模式，而研究發現，在進入到點對點之前，LINE 伺服器與手機會相互發送 UDP 封包負載長度 27 的 UDP 封包，而此 UDP 封包內容最後 6 bytes 分別為對方 IP 位址與通訊埠(如圖六)。綜上，我們將前一步驟過濾出 LINE 來的封包，以「udp.port==11000」語法過濾，即可得到通話區間；另外要取得點對點通話時對方 IP 位址的資訊，在通訊監察封包量如此龐大狀況下，以過濾語法「ip.geoip.src_asnum contains “AS38631”and frame.cap_len==69」利用每通語音通話切換至點對點通訊前的關鍵封包，將內含重要資訊取出，即可成為網路通聯。



圖六：udp 封包內含重要訊息

(2) 傳送/接收訊息：使用者進行傳送與接收訊息時，主要與 gd2.line.naver.jp、ga2.line.naver.jp 或 legy-jp-addr.line.naver.jp（IP: 203.104.153.1、203.104.153.129、203.104.150.2）進行封包交換，傳送或接收文字訊息時，可見特定封包長度的封包由伺服器傳送至通訊端，此特定長度封包長度因設備、網路環境等有所差異，常

見有 114、121、123、125 位元組等，而通訊端亦會發送特定封包長度予伺服器，形成一來一往情形，如同使用者雙方正使用 LINE 傳送文字訊息，穿插其中的封包長度就顯示雙方對話文字訊息長度。另外雙方對話同時 LINE 伺服器亦會送出一封包長度為 66 位元組之封包予通訊端，用來確認狀態，而通訊端也會發送一封包長度為 62 位元組之封包予伺服器。由 2015 年黃姓港商遭綁架勒贖案（如圖七），就可以清楚看出來歹徒正使用 LINE 通訊軟體傳收文字訊息，在案件中封包長度 114 位元組就是此一特定長度值。

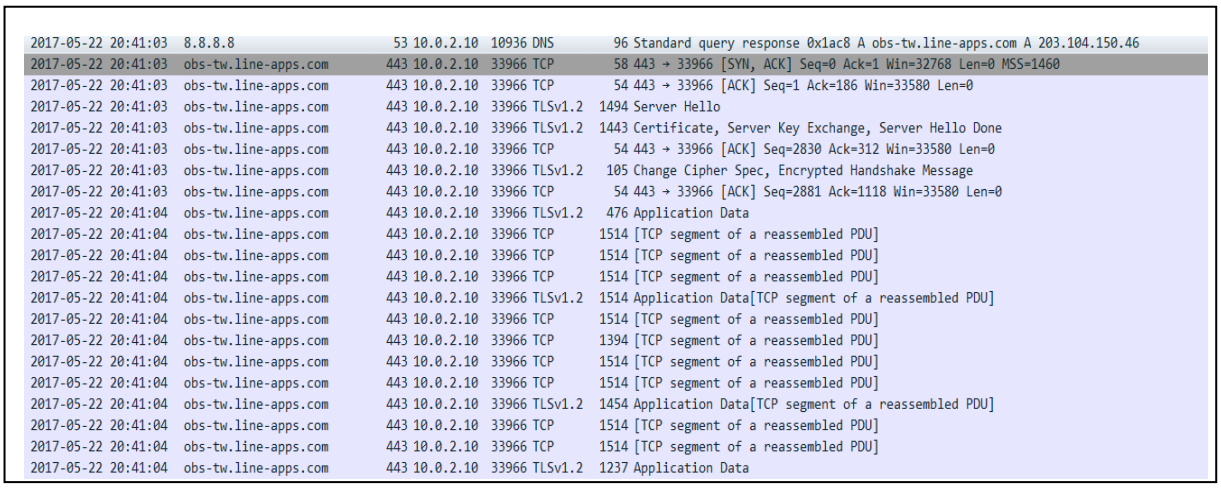
(3) 傳送接收圖片影像：使用者接收或傳送封包至伺服器 obs-jp-addr.line-apps.com、obs-tw.rts.line.naver.jp 、 obs-tw.line-apps.com （ IP:203.104.153.135 、203.104.153.134、203.104.150.4、203.104.153.7、203.104.153.6、203.104.150.46），通常圖片資料檔案較文字等為大，而每個封包大小受限（MSS 最大值 1460），因此需要將檔案切割為多個封包進行傳遞，因此在 Wireshark 中 Info 欄位會出現「TCP Segment of reassembled PDU」，但是在不同網路環境還是有些許差異，使用者若為 Wifi 無線網路環境下，傳輸內容為加密傳送（port 443），若為行動網路則無加密情形（port 80）[10]，因此在通訊監察光碟中可以看到通訊端傳送或接收之圖片或影像檔。（圖八）

(4) 加入好友：在截錄封包中如有與 lan.line.me（IP:203.104.142.52）進行金鑰交握動作者，代表使用者正進行加好友動作，主要因為 LINE 通訊軟體再傳送過程採點對點加密，因此雙方在互加好友後，會進行金鑰驗證動作。

ip.geoip.asnum contains "AS38631" and not tcp.analysis.retransmission and not tcp.len==0 and tcp

me	Source	Src Port	Destination	Dest Port	Protoco	Length	Info
2015-10-10 13:12:40	ga2.line.n…	443	10.124.189…	41738	SSL	114	Encrypted Data, Continuation Data
2015-10-10 13:12:45	10.124.189…	45375	ga2.line.n…	443	SSL	66	Encrypted Data, Continuation Data
2015-10-10 13:12:45	ga2.line.n…	443	10.124.189…	45375	SSL	114	Encrypted Data, Continuation Data
2015-10-10 13:18:21	10.124.189…	57928	ga2.line.n…	443	SSL	741	Encrypted Data, Continuation Data
2015-10-10 13:18:21	ga2.line.n…	443	10.124.189…	57928	SSL	114	Encrypted Data, Continuation Data
2015-10-10 13:18:26	10.124.189…	40996	ga2.line.n…	443	SSL	66	Encrypted Data, Continuation Data
2015-10-10 13:18:26	ga2.line.n…	443	10.124.189…	40996	SSL	114	Encrypted Data, Continuation Data
2015-10-10 13:24:15	ga2.line.n…	443	10.124.189…	54522	SSL	114	Encrypted Data, Continuation Data
2015-10-10 13:24:15	10.124.189…	54522	ga2.line.n…	443	SSL	743	Encrypted Data, Continuation Data
2015-10-10 13:24:20	ga2.line.n…	443	10.124.189…	36576	SSL	114	Encrypted Data, Continuation Data
2015-10-10 13:24:20	10.124.189…	36576	ga2.line.n…	443	SSL	66	Encrypted Data, Continuation Data
2015-10-10 13:26:15	10.124.189…	57319	ga2.line.n…	443	SSL	744	Encrypted Data, Continuation Data
2015-10-10 13:26:15	ga2.line.n…	443	10.124.189…	57319	SSL	114	Encrypted Data, Continuation Data
2015-10-10 13:26:20	10.124.189…	47342	ga2.line.n…	443	SSL	66	Encrypted Data, Continuation Data
2015-10-10 13:26:20	ga2.line.n…	443	10.124.189…	47342	SSL	114	Encrypted Data, Continuation Data
2015-10-10 13:29:03	ga2.line.n…	443	10.124.189…	51184	SSL	114	Encrypted Data, Continuation Data
2015-10-10 13:29:03	10.124.189…	51184	ga2.line.n…	443	SSL	743	Encrypted Data, Continuation Data
2015-10-10 13:29:08	10.124.189…	46045	ga2.line.n…	443	SSL	66	Encrypted Data, Continuation Data
2015-10-10 13:29:08	ga2.line.n…	443	10.124.189…	46045	SSL	114	Encrypted Data, Continuation Data
2015-10-10 13:30:56	10.124.189…	43464	ga2.line.n…	443	SSL	742	Encrypted Data, Continuation Data
2015-10-10 13:30:56	ga2.line.n…	443	10.124.189…	43464	SSL	114	Encrypted Data, Continuation Data
2015-10-10 13:32:07	10.124.189…	46791	ga2.line.n…	443	SSL	744	Encrypted Data, Continuation Data

圖七：傳送文字訊息封包



2017-05-22 20:41:03	8.8.8.8	53	10.0.2.10	10936	DNS	96	Standard query response 0x1ac8 A obs-tw.line-apps.com A 203.104.150.46
2017-05-22 20:41:03	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	58	443 → 33966 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
2017-05-22 20:41:03	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	54	443 → 33966 [ACK] Seq=1 Ack=186 Win=33580 Len=0
2017-05-22 20:41:03	obs-tw.line-apps.com	443	10.0.2.10	33966	TLSv1.2	1494	Server Hello
2017-05-22 20:41:03	obs-tw.line-apps.com	443	10.0.2.10	33966	TLSv1.2	1443	Certificate, Server Key Exchange, Server Hello Done
2017-05-22 20:41:03	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	54	443 → 33966 [ACK] Seq=2830 Ack=312 Win=33580 Len=0
2017-05-22 20:41:03	obs-tw.line-apps.com	443	10.0.2.10	33966	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
2017-05-22 20:41:03	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	54	443 → 33966 [ACK] Seq=2881 Ack=1118 Win=33580 Len=0
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TLSv1.2	476	Application Data
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	1514	[TCP segment of a reassembled PDU]
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	1514	[TCP segment of a reassembled PDU]
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	1514	[TCP segment of a reassembled PDU]
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TLSv1.2	1514	Application Data[TCP segment of a reassembled PDU]
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	1514	[TCP segment of a reassembled PDU]
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	1394	[TCP segment of a reassembled PDU]
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	1514	[TCP segment of a reassembled PDU]
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	1514	[TCP segment of a reassembled PDU]
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TLSv1.2	1454	Application Data[TCP segment of a reassembled PDU]
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	1514	[TCP segment of a reassembled PDU]
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TCP	1514	[TCP segment of a reassembled PDU]
2017-05-22 20:41:04	obs-tw.line-apps.com	443	10.0.2.10	33966	TLSv1.2	1237	Application Data

圖八：傳送圖片封包

(5) 傳送貼圖：stickershop.line-cdn.net 透過 80 埠傳送，因此聊天時傳送之貼圖未採加密方式傳送，由封包重組回來即可看到通訊端傳送之表情貼圖，惟此訊息對偵查人員並未有亟高偵查價值。

表二：Wireshark Filter 與 LINE 行為

過濾條件	BPF
過濾 LINE 相關封包並去除 ACK 及 Retransmitted 無效封包	ip.geoip.asnum contains “AS38631”and not tcp.len==0 and not tcp.analysis.retransmission
通話區間	ip.geoip.asnum contains “AS38631”and not tcp.len==0 and not tcp.analysis.retransmission udp.port==11000
通話對象 IP 位址	ip.geoip.src_asnum contains “AS38631”and frame.cap_len==69 and udp
文字訊息	ip.host contains “ga2.line.naver.jp” and not tcp.len==0 and not tcp.analysis.retransmission
傳送接收圖片時間	ip.host contains “line-apps.com”and tcp.reassembled_in
加入好友	ip.host contains “lan.line.me”

肆、結論

加密封包對偵查人員來說就像密封的信件一樣，無法由外觀窺探傳訊內容，但由信件厚度及觸感或許可以約略猜測出郵件包裹內容，本文由過去針對加密封包研究認為可將過去研究應用於目前偵辦案件所遇到之瓶頸與問題，試圖由目前封包傳遞過程中建立

每種行為模式的態樣，進一步推測使用者之網路行為。本文多以 WIFI 網路為試驗環境，使用之設備亦受限於 OPPO、Samsung、SONY 等特定型號，對於設備、網路環境等差異是否造成影響及影響範圍等均未進行實驗與評估，僅以此一概念應用於通訊監察封包分析，並冀望其他先進能更進一步引進機器學習等方式進行改進，自動對通訊監察內容進行分析，也許有機會讓案件偵查帶來一絲希望與曙光。

參考文獻

[1] S. E. Coull and K. P. Dyer, "Traffic Analysis of Encrypted Messaging Services: Apple iMessage and Beyond," *ACM SIGCOMM Comput. Commun.*, Vol. 44 , No. 5, pp.5-11, 2014.

[2] A. T. Kabakus and R. Kara, "Survey of Instant Messaging Applications Encryption Methods," *European Journal of Science and Technology*, Vol.2, No.4, pp.112-117, 2015.

[3] K. Park and H. Kim, "Encryption Is Not Enough: Inferring user activities on kakaoTalk with traffic analysis," *WISA 2015 Revised Selected Papers of the 16th International Workshop on Information Security Applications*, Vol. 9503, pp. 254-265, 2015.

[4] J. Sherry, C. Lan, R. A. Popa and S. Ratnasamy, "BlindBox: Deep Packet Inspection over Encrypted Traffic," *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* , 2015.

[5] V. F. Taylor, R. Spolaor, M. Conti and I. Martinovic, "AppScanner: Automatic Fingerprinting of Smartphone Apps From Encrypted Network Traffic, "*IEEE European Symposium on Security and Privacy*, pp.439-453, 2016.

[6] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinvic, "Robust Smartphone App Identification Via Encrypted Network Traffic Analysis," 2017.

[7] P. Velan, M. Čermák, P. Čeleda and M. Drašar, "A Survey of Methods for Encrypted Traffic Classification and Analysis," Int. J. Network Mgmt, Vol.25, pp.355-374, 2015.

[8] D. Walnycky, L. Baggili, A. Marrington, J. Moore and F. Breitinger, "Network and device forensic analysis of Android social-message applications," *Digital Investigation*, Vol 14, pp. S77-S84, 2015.

[9] F. Zhang, W. He, X. Liu and P. G. Bridges, "Inferring Users' Online Activities Through Traffic Analysis," *Proceedings of the 4th ACM Conference on Wireless Network Security*, 2011.

[10] 王傑民、伍立鈞、李泓暐、吳育松，"LINE 即時通訊軟體之通訊協定與安全性分析"，第 24 屆全國資訊安全會議，2014。